In X. Jiang, J. Hornegger, R. Koch (Eds.): Pattern Recognition. Lecture Notes in Computer Science, Vol. 8753, 629-640, Springer, Berlin, 2014. The final publication is available at link.springer.com.

A Dense Pipeline for 3D Reconstruction from Image Sequences

Timm Schneevoigt, Christopher Schroers, and Joachim Weickert

Mathematical Image Analysis Group,
Faculty of Mathematics and Computer Science,
Campus E1.7, Saarland University
66041 Saarbrücken, Germany
{schneevoigt,schroers,weickert}@mia.uni-saarland.de

Abstract. We propose a novel pipeline for 3D reconstruction from image sequences that solely relies on dense methods. At no point sparse features are required. As input we only need a sequence of color images capturing a static scene while following a continuous path. Furthermore, we assume that an intrinsic camera calibration is known. Our pipeline comprises three steps: (1) First, we jointly estimate correspondences and stereo geometry for each two consecutive images. (2) Subsequently, we connect the individual pairwise estimates and globally refine them through bundle adjustment. As a result, all camera poses are merged into a consistent global model. This allows us to create accurate depth maps. (3) Finally, these depth maps are merged using variational range image integration techniques. Experiments show that our dense pipeline is an interesting alternative to sparse approaches. It yields accurate camera poses as well as 3D reconstructions.

1 Introduction

Recovering high-quality 3D models of a static scene from images of a moving camera is an important task in computer vision. Many of the existing algorithms for this problem rely on *sparse* features. Such methods have to select the most appropriate data carefully and eliminate outliers. On the other hand, *dense* computer vision methods have made enormous progress in the last decade. For optical flow computation they belong to the leading approaches; see e.g. [1]. Moreover, dense strategies can also be on par with sparse methods for other problems such as the estimation of the fundamental matrix [16]. Dense methods do not have to put effort into selecting the best data but rather draw their robustness from using all data.

Our Goals. Motivated by these achievements, the goal of our paper is to present a pipeline for 3D reconstruction that consistently relies on dense methods: It does not require sparse features at any point. The pipeline can be divided into three stages. First we compute dense correspondences and the fundamental matrix for each consecutive image pair in a joint approach. Every pairwise estimate generally has its own scale. To estimate a consistent motion sequence these scales have to be unified. Thus, in the second step, we connect the pairwise estimates and perform a global refinement with bundle adjustment. Finally, we compute accurate depth maps and merge them into a 3D model using anisotropic range image integration.

In contrast to many existing approaches for camera motion estimation that rely on sophisticated feature descriptors, our novel pipeline uses state-of-the-art dense variational methods in every part of the process of reconstructing a 3D object from unregistered cameras. This is especially advantageous in sequences with less texture and many similar structures, where unordered feature matching is difficult.

Organization of the Paper. Our paper is organized as follows: After discussing related work in Section 2, Section 3 presents our dense reconstruction pipeline. Section 4 then evaluates its performance before the paper is concluded with a summary.

2 Related Work

Monocular tracking and range image integration are most closely related to our work. A prominent tracking approach is the PTAM algorithm by Klein and Murray [7], which tracks sparse features of the scene and creates a map of the environment. With this map the new camera pose is estimated depending on the matched features. Newcombe et al. [11] follow a similar tracking approach, but use a dense scene model instead of a sparse feature map. This way, they achieve a higher robustness to rapid motion. However, they still need standard point features to initialize their tracking algorithm. In order to describe such point features one can employ the well-known SIFT descriptor by Lowe [8], or one of the many modifications such as SURF [2] or GLOH [9].

Instead of using RGB images for tracking, also range images can be successfully employed for this task. Newcombe et al. [10] present a tracking algorithm based on dense depth frame alignment, and Izadi et al. [6] investigate a similar approach that focuses on reconstruction. Zhou et al. use points of interest to reconstruct a dense model from range images [21]. They employ a global optimization scheme which protects parts of the scene that have been scanned already. This leads to more consistent and detailed reconstructions. The disadvantage of depth sensors is their typically lower resolution compared to RGB images. Moreover, these algorithms are not applicable in certain situations such as the reconstruction from old image sequences or benchmarks where no depth data is available.

3 Our 3D Reconstruction Pipeline

This section describes the three steps of our 3D reconstruction pipeline and explains how they can be connected. First, we discuss how to obtain correspondences and epipolar geometry for all consecutive image pairs following a joint approach. Then we describe how we connect this pairwise information to compute a globally consistent camera motion with the help of bundle adjustment. Finally we explain how this enables us to construct a high quality 3D model using anisotropic range image integration.

3.1 Joint Estimation of Correspondences and Epipolar Geometry

As we assume a static scene, the only scene element that moves is the camera itself. Therefore, the moving camera can equivalently be understood as multiple identical cameras that capture the scene from different positions at the same time. In this paper, we assume that there is no knowledge about the camera movement other than that contained in the images. In order to cope with this, we propose an anisotropic modification of the joint method of Valgaerts et al. [16]. It is capable of estimating dense point correspondences and the associated fundamental matrix at the same time. Besides obtaining the fundamental matrix, this has the inherent advantage that the correspondence search is simplified as it is guided by the evolving epipolar constraint. The flow field $\boldsymbol{w} : \Omega \to \mathbb{R}^2$ over the image domain Ω between two images f_i, f_{i+1} and the fundamental matrix $\boldsymbol{F} \in \mathbb{R}^{3\times3}$ are found as a minimizer of a suitable energy in this joint approach [16]:

$$E(\boldsymbol{w}, \boldsymbol{F}) = E_D + \alpha E_S + \beta E_E \,. \tag{1}$$

Here E_D is the data term, E_S the smoothness term, and E_E denotes the epipolar term. The weights α, $\beta > 0$ balance the individual terms. In the data term we assume brightness and gradient constancy for all image points, expressed by the first and second line of the following data term, respectively:

$$\begin{aligned} E_D(\boldsymbol{w}) = \int_\Omega \Psi\big(& |f_{i+1}(\boldsymbol{x}+\boldsymbol{w}) - f_i(\boldsymbol{x})|^2 \\ & + \gamma |\boldsymbol{\nabla} f_{i+1}(\boldsymbol{x}+\boldsymbol{w}) - \boldsymbol{\nabla} f_i(\boldsymbol{x})|^2 \big) \mathrm{d}\boldsymbol{x} \,, \end{aligned} \tag{2}$$

with a sub-quadratic penalizer function $\Psi(s^2) = \sqrt{s^2 + \epsilon^2}$ and a positive weight parameter $\gamma \in \mathbb{R}$ that balances brightness and gradient constancy assumptions.

To obtain a smooth flow field $\boldsymbol{w} = (u, v)^T$, Valgaerts et al. [16] employed an isotropic flow-driven regularizer with sub-quadratic penalization. To obtain better performance, we use a flow-driven anisotropic regularizer [18]

$$E_S(\boldsymbol{\nabla w}) = \int_\Omega \mathrm{tr}\left(\Psi\left(\boldsymbol{\nabla} u \boldsymbol{\nabla} u^T + \boldsymbol{\nabla} v \boldsymbol{\nabla} v^T \right) \right) \mathrm{d}\boldsymbol{x} \,, \tag{3}$$

where tr is the matrix trace operator, and Ψ is an extension of the scalar valued function that acts on the eigenvalues of the matrix. The epipolar term directly couples the flow field $\boldsymbol{w}$ with the fundamental matrix $\boldsymbol{F}$:

$$E_E(\boldsymbol{w}, \boldsymbol{F}) = \int_\Omega \Psi \left(\begin{pmatrix} \boldsymbol{x} + \boldsymbol{w} \\ 1 \end{pmatrix}^T \boldsymbol{F} \begin{pmatrix} \boldsymbol{x} \\ 1 \end{pmatrix} \right) \mathrm{d}\boldsymbol{x} \,. \tag{4}$$

In this way it softly constrains the correspondences to fulfill an epipolar constraint. Note that the trivial solution of $\boldsymbol{F} = \boldsymbol{0}$ has to be excluded by imposing a constraint on the Frobenius norm $\|\boldsymbol{F}\|_F^2 = 1$.

As in [16], we employ the method of Lagrange multipliers to solve the constrained optimization problem (1) subject to $\|\boldsymbol{F}\|_F^2 = 1$. Thus we have to find critical points for which the functional derivatives of the Lagrangian vanish. This comes down to solving a system of equations, which in this case can be achieved by iterating between optical flow computation and fundamental matrix estimation. In the first iteration, we can simply use a zero matrix as initialisation for the fundamental matrix. This corresponds to switching off the epipolar constraint such that the fundamental matrix is recovered from pure optical flow in the first iteration.

3.2 Enforcing Scale Consistency

After processing all subsequent image pairs of the sequence as discussed in the previous subsection, we obtain fundamental matrices that describe the epipolar geometry for each image pair along with dense point correspondences (flow field). From each estimated fundamental matrix the relative pose of a canonical camera pair can be extracted [5]. Since we assume that the cameras are calibrated, i.e. their intrinsic parameters are known, the main problem at this stage is merely the remaining scale ambiguity for each pairwise estimate.

To find a mutually consistent scale we have to enforce a set of constraints that connects the whole image sequence. Each constraint has to span at least three images to contribute to a consistent scale. To build such constraints, we construct point trajectories by simply adding up flow vectors. Displacement vectors between the flow field's grid positions are estimated by bilinear interpolation. We test the validity of the resulting trajectories with a forward-backward flow consistency check. Regions of large flow gradients can originate from flow boundaries, where the estimates are less reliable. They are excluded from our computation, as proposed by Sundaram et al. [14].

However, unlike Sundaram et al., we do not limit the sampling of trajectory starting points to well textured regions. Since the idea of dense optical flow is to provide meaningful displacements even in less textured regions, we propose to spare this additional filtering effort. In our experiments, we show that a simple uniform subsampling can be used to decrease the computational effort and that this can even lead to slightly more accurate results when compared to the texture based sampling. The trajectory points should just lie on the scene object. Background regions are filtered out by given masks, which typically can be obtained by image segmentation approaches.

Even with highly accurate optical flow and despite the mentioned consistency checks, estimation and interpolation errors add up at some point. Therefore, we propose to limit the length of trajectories to a fixed value. However, short trajectories that connect only two frames may still help to improve or preserve the initially estimated relative scene geometry, even if it does not contribute to obtaining a unified scale directly.

After processing all subsequent image pairs of the sequence, we concatenate the pair-wise poses one by one to form an initial scene model. Furthermore, we triangulate the depth of one 3D point for each point trajectory to serve as an initialization. This initial scene model is then refined by enforcing the constraints imposed by the point trajectories. To this end, we minimize the reprojection error at each trajectory point by adjusting the triangulated scene point coordinates and the camera poses. This optimization procedure is known as bundle adjustment. For a detailed discussion of the theoretical concept we refer to the work of Triggs et al. [15]. We employ the implementation by Zach [20], which uses Levenberg-Marquardt steps to update the camera poses and initially estimated scene points. Since the intrinsic camera parameters are assumed to be fixed, we leave them unchanged during the optimization. We stop when either a gradient threshold or a fixed number of maximum iterations is reached. In all of our experiments we used 60 as the maximum number of iterations.

To estimate initial 3D scene points, we use the first two frames containing a trajectory thus adopting their local scale. Since badly triangulated scene points resulting from very small baselines or difficult object geometry may harm the optimization, we propose to exclude points that are triangulated to implausible depths by using a threshold. However, it is also possible to choose a different camera pair instead, in order to perform the triangulation.

3.3 Anisotropic Range Image Integration

From the optimization with bundle adjustment explained in the previous section, we have obtained a globally consistent estimation of camera motion. In return, we can now use this knowledge to compute fundamental matrices that are in accordance with the global estimation for each image pair. Then we minimize the energy from Equation 1 again, but this time we keep the fundamental matrix fixed, i.e. we only minimize w.r.t. the flow $\boldsymbol{w}$. By putting a high weight on the epipolar term, we can elegantly make use of the fact that stereo matching constitutes a 1D search problem only. This high weight allows to constrain the correspondences to closely obey the epipolar constraint. These correspondences define registered depth maps, which can be merged using variational range image integration techniques. To this end we define a 3D bounding box of interest Ω_3 and compute a signed distance field g within it for each input depth map. In the second step we find a cumulative signed distance function $u : \Omega_3 \to \mathbb{R}$ as a minimizer of the energy

$$E(u) = \int_{\Omega_3} \left(\sum_{i=1}^{N} c_i \, \Psi((u - g_i)^2) + \alpha \, S(\boldsymbol{\nabla} u) \right) \mathrm{d}\boldsymbol{x}, \tag{5}$$

Fig. 1. Sequences used for the experiments, from left to right: **(a)** Monkey. **(b)** Indy. **(c)** Plant. **(d)** Middlebury DinoRing. The Indy and Plant models are available on the *blendswap.com* platform.

where c_i are confidence weights. The unknown surface is then given by the zero level set of u. We employ the anisotropic range image integration algorithm of Schroers et al. [12]. It incorporates a direction dependent smoothing behaviour that is capable of creating smooth surfaces while preserving ridges and corners.

4 Experiments

To evaluate our approach, we compare the accuracy and stability of our estimated camera motion to a sparse feature tracker. We have selected the popular Voodoo Camera (VCT) for comparison [17]. It uses the Kanade-Lucas-Tomasi (KLT) tracker [13] and monitors the tracked image windows to detect outliers. Furthermore, it allows to perform a global refinement via bundle adjustment. Again we treat the intrinsic camera calibration as a fixed parameter for all cameras.

We test both approaches on four image sequences, which are shown in Figure 1. Three of the sequences are synthetic ones, where the camera moves around an object. The Monkey and Indy dataset consist of 112 images. The plant dataset contains 91 images. The baseline lengths between two poses vary by a factor of 2 to 3. The last sequence is the *DinoRing* dataset of the Middlebury multi-view stereo benchmark, containing 48 images.

To judge the quality of the final model one needs to overcome the gauge freedom that is basically inherent to all 3D reconstructions. Therefore, the cameras are aligned with the ground truth model before computing the individual pose errors. Each pose error consists of the Euclidean distance of the estimated camera center to its ground truth counterpart and also the difference in orientation in radians. For the orientation we do not only consider the heading but also pitch and roll.

Accuracy. Table 1 shows the errors in position and orientation of the final globally aligned model estimate for the evaluation sequences. They are being compared with the errors of the VCT. Our dense method turns out to yield

Fig. 2. Two different views of each 3D reconstruction obtained with our dense pipeline. Each view is shown with and without texture

Model	Monkey		Indy		Plant		Dino	
	$e_{\boldsymbol{c}}$	$e_{\boldsymbol{R}}$	$e_{\boldsymbol{c}}$	$e_{\boldsymbol{R}}$	$e_{\boldsymbol{c}}$	$e_{\boldsymbol{R}}$	$e_{\boldsymbol{c}}$	$e_{\boldsymbol{R}}$
Initialization	0.304	0.055	1.582	0.255	2.124	0.278	0.230	0.335
Refined	**0.012**	**0.004**	**0.341**	**0.038**	**0.705**	**0.095**	**0.007**	**0.014**
VCT (KLT)	0.013	0.005	0.437	0.054	0.809	0.105	0.010	0.019

Table 1. Quality comparison to the Voodoo Camera Tracker (VCT). The first row is the error of our initial model, showing the result when relying only on pair-wise optical flow computation. The second row is the error after the trajectory constraints have been enforced by bundle adjustment. Note that the errors in position $e_{\boldsymbol{c}}$ are not comparable between different sequences because every sequence has its own scale. Errors in rotation $e_{\boldsymbol{R}}$ are in radians.

Max	Monkey		Indy		Plant		Dino	
	$e_{\boldsymbol{c}}$	$e_{\boldsymbol{R}}$	$e_{\boldsymbol{c}}$	$e_{\boldsymbol{R}}$	$e_{\boldsymbol{c}}$	$e_{\boldsymbol{R}}$	$e_{\boldsymbol{c}}$	$e_{\boldsymbol{R}}$
2	0.310	0.055	1.501	0.285	1.701	0.246	0.194	0.344
3	0.055	0.015	1.228	0.179	1.915	0.266	0.094	0.184
8	**0.012**	0.005	0.436	0.052	1.121	0.151	0.008	0.017
16	**0.012**	**0.004**	0.341	0.038	0.705	0.095	**0.007**	**0.014**
32	0.037	0.009	**0.272**	**0.029**	**0.427**	**0.058**	0.229	0.335

Table 2. Errors of recovered camera poses for different maximum trajectory lengths.

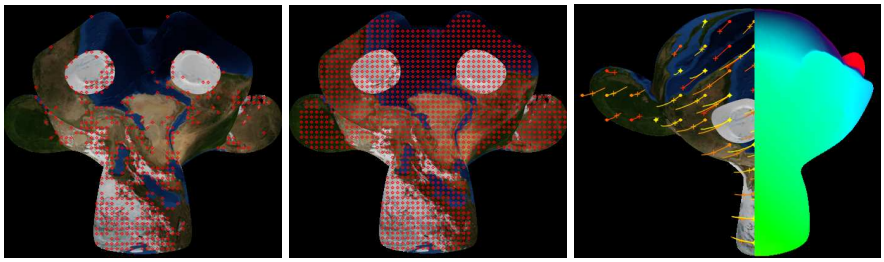
(a) Sampling Strategies

Sequence	Sampling			
	nonuniform		uniform	
	$e_{\boldsymbol{c}}$	$e_{\boldsymbol{R}}$	$e_{\boldsymbol{c}}$	$e_{\boldsymbol{R}}$
Monkey	0.015	0.005	**0.012**	**0.004**
Indy	0.379	0.041	**0.363**	**0.040**
Plant	0.753	**0.097**	**0.737**	0.098
Dino	**0.007**	**0.014**	**0.007**	**0.014**

(b) Impact of Gridsize

Grid size	$e_{\boldsymbol{c}}$	$e_{\boldsymbol{R}}$	K
2	0.307	0.035	209k
4	**0.306**	**0.034**	43k
8	0.341	0.038	10k
16	0.363	0.040	2.5k
32	0.415	0.050	0.6k
64	0.423	0.051	0.1k

Table 3. (a) Comparison of the errors of the recovered camera poses for the nonuniform sampling strategy of Sundaram et al. [14] and our uniform sampling. The best results are depicted in bold face. For uniform sampling the grid size was adjusted such that both approaches use roughly the same number of trajectories: 8k/2k/4k/9k for the respective sequence. **(b)** Results for different grid sizes using the "Indy" sequence. The number of trajectories K is given in thousands. Experiments with a grid size of 1 failed due to technical problems in simple sparse bundle adjustment (SSBA).



(a) structure based **(b)** uniform **(c)** trajectories

Fig. 3. (a) Trajectory starting points located where λ_2 of the structure tensor is above half of the average of all λ_2. **(b)** Trajectory starting points uniformly spaced on the grid. **(c)** Subset of trajectories depicted over many frames blended with the current dense flow field. Our dense flow field is accurate everywhere, which is the reason why we could show that a simple uniform sampling for trajectory starting points is preferable.

competitive results compared to VCT which relies on sparse features. Figure 2 shows the resulting 3D reconstructions for the used datasets with our dense pipeline. To obtain the textured reconstructions, we estimated the color of a point on the surface by projecting it into all cameras and then averaging the individual color values from all cameras where the corresponding surface point is visible.

4.1 Maximum Length

To find the optimal limit for trajectory lengths we list the errors of the recovered camera poses for different maximum lengths in Table 2. Basically we see that the additional information in longer trajectories improves the result, but at some point the accumulated errors can start degrading quality.

4.2 Sampling

As mentioned previously, Sundaram et al. [14] use a texture based sampling strategy for determining trajectory starting points, cf. Figure 3 (a). We employ a uniform sampling strategy instead (Figure 3 (b)). In Table 3 (a) we compare their approach to ours. The results show that ignoring the local texture measure does not decrease the accuracy of the recovered camera poses. Our more dense approach gives even slightly better results. This also holds if the grid size is adjusted such that both methods use roughly the same amount of trajectories. Table 3 (b) compares the results for different sampling grid sizes. It shows that our approach is very robust with respect to the chosen degree of density. Therefore, one may choose a coarser grid to speed up the computations. A subsampling by a factor 4 even improved the accuracy very slightly.

5 Limitations and Discussion

Bundle adjustment can lead to an undesired local minimum for an erroneous initialization. Since the main source of error in our initial model is the varying scaling between scene points triangulated from different camera pairs, it might be beneficial to only optimize for the baseline scales in a first step. This can help guiding a subsequent full bundle adjustment into a preferable minimum. A similar approach is taken in the odometry method of Fraundorfer et al. [4].

Loop closure is not directly included into our pipeline since the trajectories are initially computed over neighboring frames of an image sequence. However, after computing camera poses and the 3D reconstruction with our pipeline, it is possible to identify new frame correspondences to extend existing trajectories. This could help to solve the problem of closing loops when iterating the pipeline a second time.

Errors in the optical flow are not corrected by the bundle adjustment step, since the trajectory constraints build upon the flow field. Furthermore, the flow fields are interpolated between grid points when computing trajectories, which can introduce additional errors. However, usually this constitutes a rather small error and it is possible to limit the maximum length of trajectories. In terms of variational methods one could estimate the flow over multiple frames simultaneously. It is well-known that one can increase the flow quality by using temporal regularization; see e.g. [19].

6 Conclusions and Future Work

We have presented a novel pipeline for 3D reconstruction that completely relies on dense methods. From an image sequence of a static scene, optical flow fields and stereo geometry are jointly estimated for each consecutive image pair with a variational approach. Subsequently, the pairwise estimates are connected and globally refined through bundle adjustment. After this step, we obtain a refined model of the scene camera positions. Using these, depth maps are computed and fused by anisotropic range image integration.

Even for bad initializations our optical flow based trajectories prove to be sufficiently robust constraints, enabling the bundle adjustment to recover the correct global model. This and our comparisons to VCT show that dense approaches are an interesting alternative to sparse methods.

To be able to integrate our approach into a tracking framework for navigation purposes, future work could be based on an incremental optical flow approach, as for example discussed by Black [3]. This may speed up the estimation due to good initializations for new frames and may additionally improve the accuracy of the flow by temporal regularization.

Acknowledgements We gratefully acknowledge partial funding by the Cluster of Excellence for Multimodal Computing and Interaction.

References

1. Baker, S., Scharstein, D., Lewis, J.P., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. International Journal of Computer Vision 92(1), 1–31 (Mar 2011)
2. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) Computer Vision – ECCV 2006, Part I, Lecture Notes in Computer Science, vol. 3951, pp. 404–417. Springer, Berlin (2006)
3. Black, M.J.: Robust Incremental Optical Flow. Ph.D. thesis, Yale University, Department of Computer Science, New Haven, CT (1992)
4. Fraundorfer, F., Scaramuzza, D., Pollefeys, M.: A constricted bundle adjustment parameterization for relative scale estimation in visual odometry. In: Proc. IEEE International Conference on Robotics and Automation. pp. 1899–1904 (2010)
5. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, second edn. (2004)
6. Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., Fitzgibbon, A.: Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In: Proc. 24th Annual ACM Symposium on User Interface Software and Technology. pp. 559–568. UIST '11, ACM, New York, NY (2011)
7. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality. pp. 1–10. ISMAR '07, IEEE Computer Society, Washington, DC (2007)
8. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
9. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(10), 1615–1630 (2005)
10. Newcombe, R.A., Davison, A.J., Izadi, S., Kohli, P., Hilliges, O., Shotton, J., Molyneaux, D., Hodges, S., Kim, D., Fitzgibbon, A.: Kinectfusion: Real-time dense surface mapping and tracking. In: Proc. Tenth IEEE International Symposium on Mixed and Augmented Reality. pp. 127–136. ISMAR '11, IEEE Computer Society, Washington, DC (2011)
11. Newcombe, R.A., Lovegrove, S.J., Davison, A.J.: DTAM: Dense tracking and mapping in real-time. In: Proc. IEEE International Conference on Computer Vision. pp. 2320–2327. IEEE Computer Society, Washington, DC (2011)
12. Schroers, C., Zimmer, H., Valgaerts, L., Bruhn, A., Demetz, O., Weickert, J.: Anisotropic range image integration. In: Pinz, A., Pock, T., Bischof, H., Leberl, F. (eds.) Pattern Recognition, Lecture Notes in Computer Science, vol. 7476, pp. 73–82. Springer, Berlin (2012)
13. Shi, J., Tomasi, C.: Good features to track. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition. pp. 593–600. IEEE Computer Society, Seattle, WA (1994)
14. Sundaram, N., Brox, T., Keutzer, K.: Dense point trajectories by GPU-accelerated large displacement optical flow. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) Computer Vision – ECCV 2010. Lecture Notes in Computer Science, vol. 6311, pp. 438–451. Springer, Berlin (2010)
15. Triggs, B., McLauchlan, P., Hartley, R., Fitzgibbon, A.: Bundle adjustment - a modern synthesis. In: Proc. International Workshop on Vision Algorithms: Theory and Practice. pp. 298–375. Springer, London (2000)

16. Valgaerts, L., Bruhn, A., Mainberger, M., Weickert, J.: Dense versus sparse approaches for estimating the fundamental matrix. International Journal of Computer Vision 96(2), 212–234 (2012)
17. Viscoda: Voodoo camera tracker 1.2.0 (2012), http://www.viscoda.com/index.php/en/products/non-commercial/voodoo-camera-tracke r, accessed 10/30/13
18. Weickert, J., Schnörr, C.: A theoretical framework for convex regularizers in PDE-based computation of image motion. International Journal of Computer Vision 45(3), 245–264 (Dec 2001)
19. Weickert, J., Schnörr, C.: Variational optic flow computation with a spatio-temporal smoothness constraint. Journal of Mathematical Imaging and Vision 14(3), 245–255 (May 2001)
20. Zach, C.: Simple sparse bundle adjustment 3.0 source code (2011), http://www.inf.ethz.ch/personal/chzach/opensource.html, accessed 10/30/13
21. Zhou, Q.Y., Koltun, V.: Dense scene reconstruction with points of interest. ACM Transactions on Graphics 32(4), 112:1–112:8 (2013)